US-PAT-NO: 5857208

DOCUMENT-IDENTIFIER: US 5857208 A

TITLE: Method and apparatus for performing point in time backup operation in a
computer system

DATE-ISSUED: January 5, 1999

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Ofek; Yuval	Hopkinton	MA	N/A	N/A

ASSIGNEE INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY	TYPE CODE
EMC Corporation	Hopkinton	MA	N/A	N/A	02

APPL-NO: 08/ 656036

DATE FILED: May 31, 1996

INT-CL: [06] G06F017/30

US-CL-ISSUED: 707/204;395/182.04 ;707/202

US-CL-CURRENT: 707/204; 707/202 ; 714/6

FIELD-OF-SEARCH: 395/180; 395/181 ; 395/441 ; 395/481 ; 395/601 ; 395/620
; 395/182.04 ; 365/230.03 ; 711/200 ; 707/204 ; 707/202

REF-CITED:

U.S. PATENT DOCUMENTS

PAT-NO	ISSUE-DATE	PATENTEE-NAME	US-CL		
5133065	July 1992	Cheffetz et al.	395/181	**N/A**	N/A
5206939	April 1993	**Yanai** et al.	395/400	N/A	N/A
5208813	May 1993	Stallmo	371/10.1	**N/A**	N/A
5212784	May 1993	**Sparks**	395/575	N/A	N/A
5241668	August 1993	Eastridge et al.	395/575	**N/A**	N/A
5241670	August 1993	**Eastridge** et al.	395/180	N/A	N/A

5379398 N/A	January 1995 N/A	Cohn et al.	395/481
5379417 N/A	January 1995 N/A	**Lui** et al.	395/575
5390187 N/A	February 1995 N/A	Stallmo	371/10.1
5392244 N/A	February 1995 N/A	**Jacobson** et al.	365/200
5408644 N/A	April 1995 N/A	Schneider et al.	395/180
5430866 N/A	July 1995 N/A	**Lawrence** et al.	395/575
5435004 N/A	July 1995 N/A	Cox et al.	395/575
5473776 N/A	December 1995 N/A	**Nosaki** et al.	395/700
5487160 N/A	January 1996 N/A	Bemis et al.	395/441
5497483 N/A	March 1996 N/A	**Beardsley** et al.	395/180
5555389 N/A	September 1996 N/A	Satoh et al.	395/427
5592432 N/A	January 1997 N/A	**Vishlitzkey** et al.	395/230.03
5594900 N/A	January 1997 N/A	Cohn et al.	707/202
5613106 N/A	March 1997 N/A	**Thurman** et al.	707/204
5634052 N/A	May 1997 N/A	Morris	395/601
5649152 N/A	July 1997 N/A	**Ohran** et al.	395/441
5649196 N/A	July 1997 N/A	Woodhill et al.	395/620
5673382 N/A	September 1997 N/A	**Cannon** et al.	395/182.04
5680640 N/A	October 1997 N/A	Ofek et al.	395/839
5706467 N/A	January 1998 N/A	**Vishlitzky** et al.	365/230.03

OTHER PUBLICATIONS

Lomet et al. "Exploiting A History Database for Backup", International
Conference on Very Large Data Bases, v19, pp. 380-390, 93.

ART-UNIT: 271

PRIMARY-EXAMINER: Black; Thomas G.

ASSISTANT-EXAMINER: Rones; Charles L.

ABSTRACT:

A data network with a remote data facility for providing redundant data storage and for enabling concurrent point-in-time backup operations. A local data processing system with a data facility stores a data base and processes applications. A second system, physically separated from the first system, includes a data facility that normally **mirrors** the data in the first system. In a backup mode, the second system is enabled to transfer data from its data facility to a backup facility concurrently with, but independently of, the operation of the first system. On completion of the backup operation, the second system reconnects with and synchronizes with the first system thereby to reestablish the **mirroring** operation of the second system.

15 Claims, 6 Drawing figures

Exemplary Claim Number: 1

Number of Drawing Sheets: 6

BRIEF SUMMARY:

(1) BACKGROUND OF THE INVENTION

(2) 1. Field of the Invention

(3) This invention generally relates to backup systems for computer storage devices and more particularly to a method and apparatus for performing concurrent backups in a computer system with geographically remote redundant computer storage devices.

(4) 2. Description of Related Art

(5) Maintaining the integrity of data in computer storage devices has been and continues to be an important area of computer development. Systems today generally maintain integrity by using redundant storage devices or by using periodic backup procedures that transfer data onto a removable media. Many systems incorporate both redundancy and periodic backup procedures to benefit from the known advantages of each and to minimize the effect of the disadvantages of each.

(6) There are several ways to implement redundancy that have a

variety of names. Generally, however, the popular methods are known as RAID (Redundant Array of Independent Disks) methods that are further defined by different levels. These levels extend from a RAID-1 level in which one data storage device **mirrors** the data in another data storage device to striping in accordance with RAID-0 procedures and to variants of redundant storage of data and parity information in accordance with RAID-3 through RAID-5 procedures. These systems are all characterized by performing the corresponding redundant operation concurrently with the execution of application programs in the main system.

(7) RAID procedures are particularly useful in preventing the loss of data due to hardware failures. When a particular disk storage device fails, the data either resides on or can be reconstructed from data on other disk storage devices. However, if an event occurs, such as major damage caused by fire or the like or if an application program corrupts data, it is not possible to reconstruct the data as it existed prior to the event because redundant systems generally do not save information on an historical basis. Tape backup systems, that now also include optical disks and other media, provide a method of moving data offsite to avoid destruction as by a major physical catastrophe. They also provide an historical record because each backup generally seeks to obtain a snapshot of the entire data storage system at a particular point in time. However tape backups must be scheduled and are not made continuously.

(8) Combining both redundancy and external backups provides the potential for achieving all the advantages of the individual integrity systems and eliminating many of the disadvantages of both. However, needs of such a system have become more difficult to satisfy in recent years. For example, demands on the use or availability of the data storage devices for applications programs have increased. The size of those data storage devices has increased from capacity measured gigabytes (10.sup.9) to terabytes (10.sup.12). In computer systems with a single data storage facility, data storage devices in

the
facility or some portion of them are taken out of service during the backup
operation. In many systems the time for such backups cannot be tolerated by
the applications running on the system. Several systems that have been proposed for providing concurrent backups while avoiding these problems are
disclosed in the following U.S. Pat. Nos.:

(9) 5,212,784 (1993) Sparks

(10) 5,241,668 (1993) Eastridge et al.

(11) 5,241,670 (1993) Eastridge et al.

(12) 5,473,776 (1995) Nosaki et al.

(13) U.S. Pat. No. 5,212,784 to Sparks discloses an automated concurrent
data backup system in which a Central Processing Unit (CPU) transfers data to
and from storage devices through a primary controller. The primary controller
connects through first and second independent buses to first and second
mirrored storage devices respectively (i.e., a primary, or **mirrored** device and
a secondary or **mirroring** data storage device). A backup controller and device
connect to the secondary storage device through its bus. Normally the primary
controller writes data to both the primary and secondary data storage devices.
The CPU initiates a backup through the primary controller. In response the
primary controller then writes only to the primary data storage device and
enables the backup controller to take control of the second bus and transfer
data from the secondary data storage device to the backup media. After a
backup operation is completed, the primary controller resynchronizes the
storage devices by updating any changes that occurred to the primary data
storage device while the backup operation was underway. Examples are also
disclosed in which the primary controller connects to three and four storage
devices that enable the system to operate with redundancy by **mirroring** two
storage devices while the backup occurs with a third storage device.

(14) U.S. Pat. Nos. 5,241,668 and 5,241,670 to Eastridge et al. disclose
different aspects of concurrent backup procedures. In both systems a

request for a backup copy designates a portion of the stored data called a data set. For example, if the data storage devices contain a plurality of discrete data bases, a data set could include files associated with a corresponding data base. In a normal operation, the application program is suspended to allow the generation of an address concordance for the designated data sets. Execution of the application program then resumes. A resource manager is established to manage all input and output functions between the storage sub-systems and associated memory and temporary memory. The backup copy is formed on a scheduled and opportunistic basis by copying the designated data sets from the storage sub-systems and updating the address concordance in response to the copying. Application updates are processed during formation of the backup copy by buffering the updates, copying the affected uncopied designated data sets to a storage sub-system memory, updating the address concordance in response to the copying, and processing the updates. The designated data sets can also copy to the temporary storage memory if the number of designated data sets exceeds some threshold. The designated sets are also copied to an alternate memory from the storage sub-system, storage sub-system memory and temporary host memory utilizing the resource manager and the altered address concordance to create a specified order backup copy of the designated data sub-sets from the copied portions of the designated sub-sets without user intervention.

(15) If an abnormal event occurs requiring termination of the backup, a status indication is entered into activity tables associated with the plurality of storage sub-systems and devices in response to the initiation of the backup session. If an external condition exists that requires the backup to be interrupted, the backup copy session terminates and indications within the activity tables are reviewed to determine the status of the backup if a reset notification is raised by a storage sub-system. This enables the track extents which are active for a volume associated with a particular session to be

determined. A comparison is then made between the track events which are active and volume and track extents information associated with a physical session identification. If a match exists between the track extents which are active and the volume of and track extent information associated with a physical session identification, the backup session resumes. If the match does not exist, the backup terminates.

(16) U.S. Pat. No. 5,473,776 to Nosaki et al. discloses a concurrent backup operation in a computer system having a central processing unit and a multiple memory constituted by a plurality of memory devices for on-line storing data processed by tasks of the central processing unit. A data backup memory is provided for saving data of the multiple memory. The central processing unit performs parallel processing of user tasks and a maintenance task. The user tasks include those that write currently processed data into the multiple memory. The maintenance task stops any updating of memory devices as a part of the multiple memory and saves the data to a data backup memory.

(17) Each of the foregoing references discloses an approach for performing backup operations concurrently with the execution of applications programs in a computer system. However, in each, the system operates in the environment of a single computer system under common control. For example, in the Sparks patent the CPU connects through a primary controller to the first and second memories and to the backup controller. The Eastridge et al. and the Nosaki et al. patent references disclose systems in which the execution of applications programs is also involved in the backup operation. Further the components required for the backup operation and for maintaining redundancy are all located at a common site in each of the systems.

(18) More recently, redundancy has come to include a concept by which an array of disks at one location (i.e., a local data facility at a local site) are **mirrored** by a second array of disks at a remote location (i.e., a remote data facility at a remote site). The remote site may be in a common

building with the local site or up to hundreds of miles away from the local site. None of the foregoing systems suggest a viable solution for providing data integrity by combining redundancy and physical tape backup in such systems particularly given the apparent dependence of each of those systems on operations within the CPU that is performing applications programs.

(19) SUMMARY

(20) Therefore it is an object of this invention to provide a computer system that enables redundant storage at a remote data facility and incorporates a provision for backup into an independent media at that remote data facility.

(21) Another object of this invention is to provide a system adapted to provide backup in a remote data facility that provides a point in time backup without interfering with the operations on a data processing system at a local site.

(22) Still another object of this invention is to provide a method and apparatus for backing up data in a remote data facility that is fully transparent to operations at a local site.

(23) In accordance with this invention, first and second data processing systems at different sites are interconnected by a communications link. Each data system operates independently and includes a host computer and a data storage facility that stores data at predetermined locations in data blocks. The second system additionally includes a data backup facility. During normal operations the second system operates to **mirror** the data from the first system. A backup operation begins by isolating the first and second systems. The second system then initiates a backup operation to produce a backup of the data in the second system. Concurrently the first system records an identification of each data block in the data storage facility that changes as a result of the normal operation of the first system. When the backup operation terminates, a copy program transfers to the storage facility in the second system

data
corresponding to those data blocks in the first that were identified thereby to
reestablish the second data processing system as a **mirror** of the first data
processing system.

BRIEF DESCRIPTION OF THE DRAWINGS

It is intended that the appended claims particularly point out and distinctly claim the subject matter of this invention. The various objects,
advantages and novel features of this invention will be more fully apparent
from a reading of the following detailed description in conjunction with the
accompanying drawings in which like reference numerals refer to like parts, and
in which:

FIG. 1 is a block diagram of interconnected geographically remote data
processing systems for operating in accordance with this invention;

FIG. 2 depicts the details of a TRACK STATUS block that is useful in implementing this invention;

FIG. 3 depicts the process by which a local system as shown in FIG. 1
responds to a writing operation;

FIG. 4 depicts the process by which a remote system shown in FIG. 1 performs
a backup operation;

FIG. 5 depicts the operation of a remote link director shown in FIG. 1; and

FIG. 6 is a more detailed sequence of the remote link director operation
shown in FIG. 5.

(1) DESCRIPTION OF ILLUSTRATIVE EMBODIMENTS

(2) FIG. 1 depicts a data processing network comprising two essentially
identical data processing systems that include a local system 10 and a geographically remote system 11. A communications link 12, comprising fiber
optic cables or high-speed data transmission lines, interconnects the local
system 10 and remote system 11. The physical separation between the local
system 10 and the remote system 11 can be up to hundreds of kilometers or more.

(3) The local system 10 comprises major components including a host system 13 including a host processor and a first data storage facility that includes a system memory 14 and sets 15 and 16 of multiple data storage devices or data stores. The system memory 14 can comprise a buffer or cache memory; the storage devices in the sets 15 and 16 can comprise disk storage devices, optical storage devices and the like. The sets 15 and 16 represent an array of storage devices in any of a variety of known configurations.

(4) A channel director (CD) 17 provides communications between the host system 13 and the system memory 14; device controllers (DC) 20 and 21 provide pathways between the system memory 14 and the storage device sets 15 and 16. A bus 22 interconnects the system memory 14, the channel directors 17 and 18 and the device controllers 20 and 21. A system manager 23 enables an operator to transfer information between the various elements of the system, such as a control 24, RLD STATUS block 25 and a TRACK STATUS block 26 that are described in more detail later through one of the device controllers, namely the device controller 21 in FIG. 1. Bus access logic, not shown but known in the art, controls transfers over the bus.

(5) Generally speaking, the local system 10 operates in response to commands from one or more host systems, such as the host system 13, that a connected channel director, such as channel director 17, receives. The channel directors 17 and 18 transfer commands to a command buffer in the system memory 14. The command buffer 24 stores data structures and write requests that the device controllers generate. The device controllers, such as the device controllers 20 or 21, respond by effecting a corresponding operation using the information in the command buffer 24. The selected device controller then initiates a data operation. Reading operations transfer data from the storage devices to the system memory 14 through a corresponding device controller and subsequently transfer data from the system memory 14 to the corresponding channel director, such as channel director 17 when the host system 13 initiates the data

writing
operation.

(6) The local system 10 in FIG. 1 additionally includes a remote link director (RLD) 30 for controlling transfers of data between the local system 10
and the remote system 11 over the communication link 12. The major components
of the remote link director 30 include a control 31 and a buffer memory 32.
The! remote link director 30 connects to the system bus 22 and the communications link 12.

(7) The remote system 11 includes a remote link director 33 that connects to
the communications link 12 and includes a control 34 and a buffer memory 35.
Signals received from the remote link director 33 transfer over a system bus
36, like the system bus 22. The remote system 11, like the local system 10,
includes, as its major components, a host system 40, a system memory 41 and
storage device sets 42 and 43. The sets 42 and 43 represent an array of
storage devices configured to **mirror** the sets 15 and 16. In the same fashion
as in the local system 10, the remote system 11 includes channel directors 44
and 45 for connection to host systems. In this particular embodiment, the host
system 40 connects to the bus 36 through the channel director 44. Device
controllers 46 and 47 provide pathways between the system bus 36 and the
storage device sets or data stores 42 and 43 respectively. A system manager 50
enables an operator to transfer information between the various elements of the
system, such as a COMMAND BUFFER 51 and an RLD STATUS block 52 that are described in more detail later. Bus access logic, not shown but known in the
art, controls transfers over the bus.

(8) Each of the local and remote systems 10 and 11 may comprise a Symmetrix
integrated cached disk array as manufactured and sold by the assignee of this
invention according to known operations as described in Yanai et al., U.S.
Pat. No. 5,206,939 issued Apr. 27, 1993. Consequently, the following discussion makes only general references to general operation of such a systems. For purposes of this invention it is sufficient to understand that
the remote system 11 normally acts as a **mirror** of the local system 10 on a

volume-by-volume basis and that the volume can be physical volumes, although **logical volumes** are preferred. Given the geographical separation between the local and remote systems 10 and 11, the system in FIG. 1 operates with an extremely high degree of reliability, even in the event of a natural disaster. Normally, the local system 10 is the active system while the remote system 11 acts as a **mirror**. In such systems transfers from the local system 10 to the remote system 11 normally occur in response to a writing command issued by a local host system such as the host system 13. The details of such a transfer are discussed later.

(9) The host system 40, in such an environment, typically will be limited to performing read operations in order that the remote system 11 exactly **mirror** the local system 10. Should some catastrophic event prevent any part of the local system 10 from operating, control can be transferred to the remote system 11 through use of the system manager 50 whereby the remote link director 33 effectively disconnects from the local system 10 to enable the host system 40 to read and write data to the storage device sets 42 and 43. **Mirroring** remote data facilities are also known in the art and Symmetrix remote data facilities supplied by the assignee of this invention provide such remote **mirroring** capabilities.

(10) Unlike the prior art operation of the local and remote systems like those shown in FIG. 1, a system constructed in accordance with this invention enables the remote system 11 (1) to disconnect from the local system 10, (2) to enable all the data to transfer to a conventional backup unit 53, such as a conventional tape backup unit, (3) to reconnect to the local system 10 and (4) to resynchronize to the local system 10 and remote system 11 automatically.

(11) This operation requires two types of information, namely: the status of the remote link directories 30 and 33 and the status of each track or corresponding data block in storage device sets 42 and 43. The RLD STATUS

block 25 records the status of the remote link directory 30. For purposes of this discussion, it is assumed that the RLD STATUS block 25 has one of three values that represent a "DISCONNECT FOR BACKUP" or "BACKUP" status, a "BACKUP RETURN" status and an "ONGOING" or normal operating mode status. The BACKUP status value indicates that an operator at the local system 10 or the remote system 11 has utilized the corresponding one of the system managers 23 and 50 to terminate communications between the local system 10 and the remote system 11 for the purpose of performing a backup. The RETURNING status means that the system manager 23 or 50 has just reestablished the communications. During intervals characterized by the "BACKUP" and "RETURNING" status, the remote system 11 does not **mirror** the local system 10. The ONGOING status means that the local system 10 and the remote system 11 are operating normally and are synchronized.

(12) The TRACK STATUS block 26 comprises a bit map with an entry for each track or data block on the storage device sets 15 and 16. FIG. 2 represents the TRACK STATUS block 26 as a matrix in which each row identifies a track in the storage device sets 15 and 16 and in the sets 42 and 43. In FIG. 2 the columns are headed by M1, M2, M3 and M4 that establish a correspondence between the column position and the system containing the TRACK STATUS block in the local system 10 and in each of up to three **mirroring** systems.

(13) It will be apparent that each entry in the block 26 corresponds to a data block of a minimum transfer size. In Symmetrix systems this is typically a track; however, a given track may be divided into multiple blocks or a block might even comprise multiple contiguous tracks. Such variations only change the track status block 26 by increasing or decreasing the number of rows in the TRACK STATUS block 26, as each row will correspond to one data block.

(14) In the system of FIG. 1, only the data columns identified as the M1 and M2 columns contain relevant TRACK STATUS data as only one local system 10 and

one remote system 11 are present. For any given track the M1 column in FIG. 2 indicates whether the data in the corresponding track in the local system 10 is valid while the M2 column indicates whether the data in the corresponding track in the remote system 11 is valid. In an implementation involving two additional remote systems, the M3 and M4 columns in FIG. 2 would indicate the whether the data in the corresponding tracks in the remaining two **mirrored** systems were valid. Typically and for purposes of this discussion, a "0" indicates a valid data track or block; a "1", an invalid data track or block.

(15) With this as background, it will now be possible to describe the various operations of these components (1) during a normal **mirroring** mode, (2) during a backup mode and (3) during the return to a normal operating mode.

(16) NORMAL **MIRRORING** MODE

(17) In a normal operating mode the local system 10 is the active system while the remote system 11 functions solely as a **mirror**. For example, when the system in FIG. 1 accommodates a database, the local system 10 generally processes applications including those that can effect changes to the data base. For purposes of this description, it is assumed that the host system 13 issues a Channel Control Word (CCW) command including all the necessary parameters from which the system can transfer a data block to or from a particular location in the storage device sets 15 and 16. Other operating systems use other procedures. However, this invention is readily adapted to operate with such systems.

(18) When a host system such as the host system 13 in FIG. 1 issues a command, it transfers the CCW command or equivalent to the channel director 17 for transfer to the system memory 14. If the system memory control 24 determines that the pending CCW command will perform an operation other than a writing operation for transferring data to a location in one of the storage device sets 15 or 16, the control 24, in step 60 of FIG. 3, diverts to perform the requested operation in step 61. If the CCW request defines a write operation, control transfers from step 60 to step 62 wherein the information is written into the system memory 14 for subsequent transfer to locations in the

storage device sets 15 and 16 in a normal fashion.

(19) During normal **mirroring** operations, the RLD STATUS block 25 indicates an ONGOING status because the remote system 11 connects to the local system 10 through the remote link directors 30 and 33 and the communications link 12 and because the local system 10 and remote system 11 are synchronized. Consequently control transfers from step 63 in FIG. 3 to step 64 where the system awaits an acknowledgement signal that the remote system 11 has received the data being written to its system memory 41. When this acknowledgement is received under predetermined constraints, control transfers to step 65 wherein the control 24 sends a CE, or Channel End, signal to the host system 13 in step 65. If this is the first or an intermediate CCW command in a sequence, step 66 transfers control to step 67 to send a DE, or Device End, signal to the host system 13. After processing the last CCW command in a sequence step 66 diverts to step 70 to test for any error conditions. If no error has occurred, step 67 sends the DE signal to the host system 13. If an error occurred, control passes to step 71, and the control 24 transfers the DE signal with a message identifying the nature of the error.

(20) Consequently during the normal operating mode any changes the host system 13 makes to the data in the storage device sets 15 and 16 automatically produce corresponding changes in the storage device sets 42 and 43. In normal operation the storage device sets 42 and 43 or **logical volumes** therein exactly **mirror** the corresponding ones of the storage device sets 15 and 16 or **logical volumes** therein according to configuration information from the system manager 23 and system manager 50. Although the host system 40 is enabled to access data in the storage device sets 42 and 43 in this mode, it can not alter data. It can access data only on a read-only basis. In the normal operating mode and in the context of a data base system, the local system 10 processes on-line transaction processing applications by altering the storage device sets 15 and 16 that constitute a primary repository for the data base. It may also

process
decision support system applications. The remote system 11 normally operates
only as the **mirror** of that data base.

(21) BACKUP MODE

(22) In accordance with this invention, it is possible for the host system 40
in FIG. 1 to operate independently with the capability of reading information
to the storage device sets 42 and 43 and of transferring that information to
the backup unit 53. A backup operation begins by using the system manager 50
to block communications through the remote link directors 30 and 33 and communications link 12. Well known processes then update the RLD status
registers 25 and 52 in the local system 10 and remote system 11, respectively
by shifting the status from the "NORMAL" operating mode to the "BACKUP" mode
and altering the operations within the local system 10 and the remote system 11
differently.

(23) Referring again to FIG. 3, any writing operation or updating operation
that occurs in the local system 10 during the BACKUP operating mode still
alters data in the storage device sets 15 and 16 in step 62 in FIG. 3. However, in step 63 the control 24 determines that the remote system 11 is
disconnected because the RLD STATUS block contains the "BACKUP" status. In
step 72 the control 24 updates the corresponding TRACK STATUS block 26 to
indicate that the remote system 11 no longer contains valid data in the corresponding track because it is not possible to transfer the new data to the
remote system 11. In the system of FIG. 1 the corresponding register on the
block 26 would be set to "01" for the M1 and M2 sets. The operation of step 72
also occurs if step 73 indicates that a time interval has elapsed without the
receipt of an acknowledgement signal, during the normal operating mode.

(24) Thus during the backup mode the host system 13 continues on an uninterrupted basis to process various applications on the data base or other
data collection in the storage device sets 15 and 16. This occurs with no
significant increase in the time required because the only additional requirement is to set the "M2" bit in the corresponding entry of the TRACK

STATUS block 26 to an invalid state (e.g., a "1") in step 72 and because the control 24 performs this function.

(25) Once the communications link 13 has been disabled, the remote system 11 responds according to FIG. 4. In step 80 the host 40 is enabled to issue CCW commands that implement a backup operation. Step 81 determines that in fact the system is operating in the BACKUP mode. If not, the control 51 diverts its activities to step 82 to initiate an appropriate error or other procedure. Otherwise in step 83 the control 51 bit begins the backup operation to produce a "point-in-time" backup, the time being the instant the system manager disables transfers. The host processor 40 in FIG. 1 controls the backup unit 53 in this particular embodiment. Generally the host processor will issue a series of commands to read files in succession, although other reading sequences, as track-by-track, could be substituted.

(26) These are conventional read commands that, in a Symmetrix unit, initially attempts to read data in the system memory 41. If not successful, the control 51 transfers the requested data from the address locations in the storage device sets 42 and 43 to the system memory 41.

(27) The backup operation continues until step 84 determines that all data has been transferred. That backup may, of course, include all the data or selected portions (e.g., files). Upon completion, step 84 diverts to step 85 to determine whether any errors occurred. If no error occurs, step 86 signals the end of the backup operation so the host system 40 can reenable the path to the local system 10. If an error occurs step 87 produces the signal with an appropriate error identification message. Thus, during this backup mode, the host system 40 transfers all the selected data from the storage device sets 42 and 43 to the backup unit 53.

(28) FIG. 5 depicts the pertinent operation of the remote link director 30 at the local system. The control 31 in step 90 determines whether the path through the communications link 12 to the remote link director 33 is effective. If it is not, the control 31 can set the RLD status to the "BACKUP" status in

step 91 merely to provide an interval before step 90 tests the status again.
Once the path is disabled, the status remains unchanged until a reconnection at the end of the backup mode.

(29) RETURN TO NORMAL OPERATING MODE

(30) When the backup concludes, the system manager 50 reestablishes the connection through the communications link 12 and reverts the remote system 11 to the normal operating mode. Simultaneously the control 31 shifts control from step 90 in FIG. 5 to step 92 and determines whether the connection is being made after the remote system has operated in an backup mode based upon information contained in the RLD STATUS block 25 or any alternate location within the remote link director 30. If the two remote link directors 30 and 33 have disconnected for other reasons, then step 92 transfers to step 93 to signal an error condition or take any other appropriate action. Otherwise, the control 31 sets the RLD STATUS block 25 to a "BACKUP RETURN" status in step 94 to indicate a return to the normal operating mode during which resynchronization will occur. Then in step 95 the control 31 resynchronizes the local system 10 and remote system 11. Generally, the control 31 retrieves the TRACK STATUS block 26 and identifies all the tracks in the storage device sets 42 and 43 that have invalid tracks because the host system 13 altered tracks in the data storage sets 15 and 16.

(31) In one embodiment of this invention, the control 31 performs the resynchronization process of step 95 according to a procedure of FIG. 6. Before discussing this procedure in detail, it will be helpful to understand that at the end of the independent operating mode the collection of bits assigned to a specific track in the TRACK STATUS block 26 and assigned to the local system 10 and **mirroring** remote system 11 can define only one of two valid bit patterns, namely M1=0 and M2=0 or M1=1 and M2=1 or "00" or "01". That is, if the host system 10 does not alter information in a track during the backup mode, the corresponding M1 and M2 bits in the TRACK STATUS block 26 will be

of the
M1 and M2 bits will be "01" indicating that the data on the track in the local
system is valid, but that the data in the corresponding track of the remote
system 11 is invalid.

(32) FIG. 6 depicts the process by which the control 31 in FIG. 1 uses these
bit patterns to resynchronize the systems. This process is iterative in nature
and under the control of a loop controller in the form of a track counter (not
shown, but located within the RLD 30) that the process initializes in step 100.
In step 101 the control 31 forms a vector corresponding to the data from the
TRACK STATUS block 26 for the local system 10 and the remote system 11 that
performed the backup.

(33) In step 102, the control 31 determines if the vector has a "ZERO" value,
as would occur if no change had occurred in the local system 10. In that
event, control passes to a loop control comprising step 103 that increments the
track counter to point to a next track in sequence. In step 104 the control
determines if all the tracks have been tested by comparing the track counter
contents to a maximum value. If more tracks need to be examined, control
passes back to step 101. Otherwise the resynchronizing process is complete,
and step 104 transfers control to step 105 to restore the status in the RLD
STATUS block to the "ONGOING" value indicating a return to normal **mirroring**
operations.

(34) If the vector does not have a "ZERO" value, the control 31 transfers
from step 102 to step 106. If the value of the vector is other than "01", then
an error exists. The control 31 terminates any further processing with respect
to the particular track by noting the error in step 107 through an error
condition detection scheme or interrupt handler and then transfers to step 103
in the loop control.

(35) If the vector has a value of "01", the tracks need to be resynchronized.
Step 106 then transfers to step 110 to copy the track from the local

system 10
to the remote system 11. Next the system transfers operations to step 103 in
the loop control.

(36) When step 104 shifts control to step 105, the resynchronizing process of
FIG. 6 has tested the bit patterns for each track and copied only those that
are needed to resynchronize the data. This operation occurs concurrently with
normal operations so that during the process any changes the host system 13
makes to the data also produces a change in the remote system 11. If the host
system 13 alters a track during the process, the new data transfers to the
remote system 11 conventionally. If the host system 13 alters the track before
it is processed by the resynchronizing process the copy program 97 will merely
recopy the data from the local system 10 to the remote system 11.

(37) As previously indicated it is possible to modify the network shown in
FIG. 1 by adding a third and even a fourth system interconnected through
corresponding communications links. The interconnection of three systems could
then provide a first system like the local system 10 dedicated to process OLTP
or other priority applications, a second remote system like the remote system
11 operating as a **mirror** and as a mechanism for performing point-in-time
backups, and a third system that always operates to provide a second **mirror** of
the data in the first system. Alternatively, the third system could also be
adapted for running other applications.

(38) The general approach of redundancy and remote backups of this invention
is particularly effective because the percentage of operations that alter the
data on a disk rarely involve the system for a majority of its time. Normally,
significantly less then half of all disk operations involve writing operations
or data changes. Further the remote system can operate as a backup facility
because generally such backups are taken of a snapshot of the data base taken
at a particular time. In this particular embodiment that snapshot represents
the data base at the instant the system manager 50 disables transfers

through
the communications link 12.

(39) When implemented as described above, the network shown in FIG. 1 meets the objectives of this invention. The local system 10 and the remote system 11 operate in a **mirrored** configuration for the vast majority of time to provide redundancy. However, when it is necessary to obtain a backup, that operation occurs at the remote system 11 concurrently with the continued operations within the local system 10 and without any intervention by the local system 10 that could adversely affect its operating characteristics. Immediately upon completion of the backup, the local and remote systems resynchronize to reestablish a **mirror** relationship. Typically the number of tracks that need to be updated will be minimal, so that the time required to resynchronize the system after running decision support system applications will be minimal. Moreover the copy program, by virtue of its being located in the remote link director 30, performs this resynchronization independently of the processing of applications on the local system 10.

(40) This invention has been disclosed in terms of an embodiment based upon the architecture of the assignees Symmetrix data facilities. Specific implementations are therefore system specific. Discussion of other particular implementations have not been incorporated. Rather the discussion has been directed to how these different systems interact for implementing the remote point-in-time backup concept of this invention and provide sufficient information for enabling an implementation on the data processing systems of other manufacturers.

(41) In this specific embodiment, data transfers occur on a track-by-track basis with the monitoring of the status of those tracks in the TRACK STATUS block 26 of FIGS. 1 and 2. Other embodiments might operate by transferring data blocks of a longer or shorter length than is carried on a single track. In such an implementation, the TRACK STATUS block 26 would be modified to identify each such block individually. Moreover, the system in FIG. 1 depicts

a single host system 13 in the local system 10 and a single host system 40 in
the remote system 11. Other systems like the remote system 11 could connect to
the local system 10 by separate remote link detectors and communications links.
In such a configuration, each remote system could **mirror** the entirety of the
data or portions of the data in the device storage sets 15 and 16. In other
embodiments, two or three systems, like the local system 10 could connect to
the remote system 11 by means of separate remote link directors and
communications links whereby the capacity of the disk storage sets 42 and 43
would be increased to equal all the disk storage capacity to be **mirrored**
collectively in the remaining systems. It will also be apparent other host
systems could be added to the configuration in FIG. 1 as by being connected to
the channel director 17 or other channel directors, such as channel director
18. It will be apparent that many other modifications can be made to the
disclosed apparatus without departing from the invention. Therefore, it is the
intent of the appended claims to cover all such variations and modifications as
come within the true spirit and scope of this invention.

CLAIMS:

What is claimed as new and desired to be secured by Letters Patent of the
United States is:

1. In a data network including a first data processing system with a first
host computer and a first data storage facility for processing application
programs, including a second data processing system with a second host
computer, a second data storage facility and a backup facility for providing a
copy of the data in the second data storage facility on separate media, and
including a communications link for interconnecting the first and second data
processing systems, the second data processing system operating in a normal
operating mode to **mirror** the first data storage facility by transferring data
through the communications link and wherein each of the data storage facilities
includes at least one disk storage device characterized by a plurality of

tracks and by block data transfers having a one-track length, the improvement of a method for enabling the **backup of the data in the network** without interfering with the operation of the first data processing system, said method comprising the steps of:

(A) enabling from the second data processing system a backup operating mode by disabling transfers over the communications link and initiating a backup operation from the second data storage facility to the backup facility, the backup operation occurring in parallel with operations in the first data processing system

(B) recording, in the first data processing system, each transfer to the first data storage facility during the backup operating mode produced the first data processing system,

(C) converting, upon return from the backup operating mode to a normal operating mode, the recordings of each transfer into a list of changed tracks, and

(D) copying the data from each changed track of the first storage facility identified by the list of changed tracks to each corresponding track in the second date storage facility.

2. A method as recited in claim 1 wherein the communications link includes a link director in each of the first and second data processing systems for controlling transfers over the communications link and wherein the link director in the first data processing system performs said copying concurrently with the operation of the first data processing system.

3. A method as recited in claim 1 the first data processing system maintains, for each track in the first data storage facility, track status defined by a first field indicating the validity of the track in the first data processing system data storage facility and a second field indicating the validity of the corresponding track in the second data processing system data storage facility and wherein the recording of track identifications in the first data processing system includes responding to each change in the

data on
a track by setting the second field in the track status for the
corresponding
track to a value for invalid data.

4. A method for operating first and second data processing systems interconnected by a communications link, each data processing system being
capable of independent operation and including a host computer and a data
storage facility that stores data at predetermined locations in data blocks,
the second data processing system including a data backup facility and,
operating, during a normal operating mode, to **mirror** in the data storage
facility of the second data processing system the data in the data storage
facility of the first data processing system in response to a copy program,
said method producing a point-in-time backup on the data backup facility and
comprising the steps of:

(A) disabling the copy program thereby isolating the first and second data
processing systems and enabling the first data processing system to continue
its operations,

(B) initiating the operation of the backup facility at the second data
processing system thereby producing a backup of the data in the data storage
facility of the second data processing system,

(C) recording, at the first data processing system and during the backup
operation, an identification of each data block in the data storage means of
the first data processing system that changes as a result of the operation of
the first data processing system, and

(D) enabling the copy program upon completion of the backup operation
thereby copying data blocks from the data storage facility in the first data
processing system to the data storage facility in the second data processing
system corresponding to the recorded identifications in the first data
processing system thereby reestablishing the second data processing system as a
mirror of the first data processing system.

5. A method as recited in claim 4 wherein the communications link includes

a first link director connected to the first data processing system and a
second link director connected to the second data processing system and wherein
the first link director performs said copying and wherein said copying occurs
concurrently with operation of the first data processing system after the
normal operating mode is established.

6. A method as recited in claim 5 wherein the first data processing system
maintains data block validity status that, for each data block, includes a
first field indicating the validity of the data block in the first data
processing system data storage facility and a second field indicating the
validity of the data block in the second data processing system data storage
facility and wherein the recording of data block identifications in the first
data processing system includes the step of responding to each change in a data
block produced by the first data processing system by setting the second field
in the corresponding data block validity status to a value that indicates
invalid data.

7. A method as recited in claim 6 wherein each of the data block status
fields comprises a single bit having first and second states when the
corresponding data is valid and invalid, respectively, and wherein said copying
step includes converting the status of the second bits at the second state into
a list of data blocks that had been changed by the first data processing system
during the backup operation.

8. In a data processing network including first and second data processing
systems interconnected by a communications link, each System being capable of
independent operation and including a host computer and a data storage facility
that stores data at predetermined locations in data blocks, said second data
processing system additionally including a backup facility and, during a normal
operating mode, operating to **mirror** the data in said data storage facility of
said first data processing system, the improvement of:

(A) mode control means in said second data processing system for establishing the normal operating mode and for establishing a backup

operating
mode by disabling transfers through said communications link and
enabling said
backup facility in said second data processing system and enabling said
first
data processing system to continue its operations,

(B) recording means in said first data processing system for
recording an
identification of each data block in said data storage means of said
first data
processing system that changes as a result of the operation of said
first data
processing system,

(C) copying means at said communications link for copying data
blocks from
said data storage facility in said first data processing system to the
data
storage facility in said second data processing system, the data blocks
corresponding to the recorded identifications in said first data
processing
system after said mode control means reestablishes the normal operating
mode
thereby to reestablish said second data processing system as a **mirror**
of said
first data processing system.

9. A network as recited in claim 8 wherein said first data
processing
system includes, for each data block, a data block status register
means
defined by a first field indicating the validity of the data block in
said
first data processing system data storage facility and a second field
indicating the validity of the data block in said second data
processing system
data storage facility, said recording means responds to each change in
a data
block produced by said first data processing system by setting said
second
field in said corresponding data block validity status to a value
indicating
invalid data.

10. A network as recited in claim 9 wherein said copying means
includes
means for generating a changed track list in response to the track
status
registers with the second fields indicating invalid data thereby to
identify
all data blocks in said second data processing system that fail to
mirror
corresponding blocks in said first data processing system, said copying
means
being responsive to said changed track list by copying each of said

identified
data blocks from said data storage facility of said first data
processing
system to said data storage facility of said second data processing
system.

11. A network as recited in claim 10 wherein each of said data
block status
register fields comprises a single bit having first and second states
when the
corresponding data is valid and invalid, respectively.

12. In a data network including a first data processing system with
a first
host computer and a first data storage facility for processing
application
programs, including a second data processing system with a second host
computer, a second data storage facility and a backup facility for
providing a
backup copy of data in the second data storage facility on separate
storage
media and including a communications link for interconnecting said
first and
second data processing systems for normal operation wherein said second
data
processing system operates to **mirror** said first data storage facility
by
transferring data through said communications link and wherein each of
said
data storage facilities includes at least one disk storage device
characterized
by a plurality of tracks and by block data transfers having a one-track
length,
the improvement of a method whereby said backup facility is enabled to
backup
the data in the network without interfering with the operation of the
first
data processing system, the improvement comprising:

(A) mode control means in said second data processing system for
establishing the normal operating mode and for establishing a backup
mode for
enabling said backup facility in said second data processing system to
backup
data in said second data storage facility while enabling the first data
processing system to continue its operations,

(B) status registers in said first data processing system for
recording each
transfer to said first data storage facility during the backup
operating mode
produced by said applications programs in said first data processing
system,

(C) means in said first data processing system for converting, upon
return

to a normal operating mode, the recordings of each transfer into a changed track list, and

(D) means for thereafter copying to each track in said second data storage facility the data from each track of said first storage facility identified by said changed track list.

13. A network as recited in claim 12 wherein the communications link includes a link director in each of said first and second data processing systems for controlling communications through said communications link and wherein said copying means operates in said link director in said first data processing system concurrently with the operation of said first data processing system.

14. A network as recited in claim 13 wherein said each track status register in said first data processing system includes a first field indicating the validity of data in a corresponding track in said first data processing system data storage facility and a second field indicating the validity of the data in the corresponding track in said second data processing system data storage facility, changes in the data of a track produced by an application program causing the setting of said second field in said corresponding track status register to a value indicating invalid data.

15. A network as recited in claim 14 wherein each of said track status register fields comprises a single bit having first and second states when the corresponding data is valid and invalid, respectively, said converting means producing the changed track list in response to track status registers in which the second field bit indicates a track in the second data storage facility with invalid data.

US-PAT-NO: 6490598

DOCUMENT-IDENTIFIER: US 6490598 B1

TITLE: System and method for external backup and restore for a computer data
storage system

DATE-ISSUED: December 3, 2002

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Taylor; Kenneth J.	Franklin	MA	N/A	N/A

ASSIGNEE INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY	TYPE CODE
EMC Corporation	Hopkinton	MA	N/A	N/A	02

APPL-NO: 09/ 466976

DATE FILED: December 20, 1999

INT-CL: [07] G06F017/30

US-CL-ISSUED: 707/204;707/202 ;345/356

US-CL-CURRENT: 707/204; 707/202

FIELD-OF-SEARCH: 707/202; 707/204 ; 345/356

REF-CITED:

U.S. PATENT DOCUMENTS

PAT-NO	ISSUE-DATE	PATENTEE-NAME	US-CL		
6047294	April 2000	Deshayes et al.	707/204	**N/A**	N/A
6092086	July 2000	**Martin** et al.	707/202	N/A	N/A
6199074	March 2001	Kern et al.	707/204	**N/A**	N/A
6329985	December 2001	**Tamer** et al.	345/356	N/A	N/A
2002/0002661	January 2002	Blumenau et al.	711/165	**N/A**	N/A

OTHER PUBLICATIONS

"Integration of EMC TimeFinder and SRDF Software Within Informix Dynamic Server,"p/n AB51815, Engineering White Paper, EMC Corporation, Dec. 12, 2000.*

EMC Data Manager Product Description Guide, 1999, Chapter 4, EDM Symmetrix Connect and EDM Symmetrix Path, pp. 23-31.

ART-UNIT: 2175

PRIMARY-EXAMINER: Rones; Charles L.

ABSTRACT:

A system providing intelligent, integrated external backup and restore for databases and DBMS (data base management systems) which are stored on data storage systems. An interface between the data storage system backup system and the DBMS allows the backup system to work with the DBMS for greatly enhanced backup and restore. External backup and restore frees up the host systems from having to pipeline the data from the data storage system to the backup system. The backup system is able to determine which files stored in the data storage system should be backed up, based on querying the DBMS. This invention is useful for systems including RAID data storage system storing databases (including relational and object oriented), and provides for partial as well as complete backup and restore options.

8 Claims, 4 Drawing figures

Exemplary Claim Number: 1

Number of Drawing Sheets: 4

BRIEF SUMMARY:

(1) FIELD OF THE INVENTION

(2) This invention is directed towards data storage systems, and more particularly towards physical backup and restore of databases residing in data storage systems.

(3) BACKGROUND

(4) Computer systems allow the processing of massive quantities of data for a variety of purposes. As the ability to process data has increased, so has the

need for data storage systems which provide massive data storage capabilities combined with fast access for host systems. Another feature required by many businesses and industries is continuous availability. Many businesses operate on a world-wide basis, and have a need for round-the-clock access to databases stored in one or more data storage systems. The data stored in these data storage systems is changing at an incredible rate, for example with transaction processing, reservation systems and data mining, the data is changing and updating many times per second.

(5) Another requirement for data storage systems is periodic backup of data both for archival purposes and for data recovery in case of a system failure. For many businesses, a loss of data can be catastrophic. Therefore, system backups must be performed on a frequent basis.

(6) However, the need for system backups often interferes with the need for continuous availability. With many data storage systems, performing a system backup requires taking the data storage system offline, thereby denying continuous access to the data.

(7) One solution to this problem is used for RAID (Redundant Array of Independent Disks) systems. In RAID-1 systems, two physical storage devices, such as disks, each store identical data, in a process known as "**mirroring**". This provides a very high level of fault tolerance in the form of redundancy, and it also allows data backups to be performed while still allowing continuous data access. Typically, the **mirroring** process is stopped (referred to as splitting the **mirrors**), and one of the disks is taken off-line and backed up, while the other disk remains online and available. When the first is completely backed up, the two are resynchronized (so that the data is identical on both), and the data storage system returns to full operation.

(8) However, there are still problems related to backing up at the physical disk volume level, instead of at the application level. For example, a database (whether hierarchical, relational, object-oriented or otherwise) stores data in a logical structure which does not match physical disk details.

Several layers of mapping are performed to map the database data onto the physical disks. Modem data storage systems perform mapping of physical disks to **logical volumes,** to support a standard representation of storage units to host systems. These **logical volumes** appear to host systems as a defined set of storage volumes for the host to access. The data storage systems perform all functions of converting **logical volume** addressing and accessing to functions effective on physical disks.

(9) But there is still one layer of mapping (or more) from the database application to **logical volumes**. The database application performs much of this mapping function, to provide the host systems with access to the database data in a format dictated by the database application. The host systems benefit from having all the mapping details performed automatically by the database applications and the data storage systems. However, the layers of mapping make the process of backup and restore much more difficult. Traditional backup systems are unable to back up the variety of data formats and client platforms that exist in heterogeneous, growing networks, leaving potential holes in **network backup** coverage and leaving administration decentralized. There is little integration of the backup systems to allow "intelligent" backups by taking advantage of the mapping layers, for example to perform incremental backups. Often, the only solution is for "brute force" complete database space backups, which are inefficient and time consuming. This causes a great discrepancy between the advantages which many database applications provide to host systems. Ultimately, there is significant difficulty in performing timely data storage system backups as necessary to prevent catastrophic data loss.

(10) Some presently available database applications have a non-integrated approach for external backup of **mirrored** database spaces. This approach blocks the data storage system server, allowing only read-only accesses to run, therefore it is only usable for a **mirrored** data storage system as previously

described. Further, this approach requires blocking the data storage system server for the length of time necessary to split **mirrors**. Also, the user must manually perform logical restore to successfully restore their spaces. These non-integrated approaches essentially involve making the DBMS accept on faith that the user has done a physical restore. There is great risk in depending on the users to perform all backup/restore procedures correctly.

(11) SUMMARY

(12) The present invention provides for intelligent, integrated external backup and restore for DBMS which is stored on RAID data storage systems.

(13) According to the present invention, an interface between the data storage system backup system and the DBMS allows the backup system to work with the DBMS for greatly enhanced backup and restore. External backups can be created by blocking the server after forcing a checkpoint, whereupon the backup is created using an external resource. In one embodiment, this is accomplished by disconnecting the **mirrored** target data from the source online data, so that the target data can be saved to a safer site other than the production site. After the backup is created, the server is unblocked to resume normal server operations.

(14) An example of a relational database is the Informix 7.3 database and database management system (DBMS) from Informix Software Inc. of Menlo Park, Calif. The Informix DBMS runs on several Unix platforms as well as Microsoft Windows.RTM.. The Informix DBMS provides very limited access to other applications attempting to perform system utility operations such as backup/restore. An illustrative embodiment of the present invention provides for external backup/restore of Informix 7.3 database spaces and blob spaces, through an interface to the Informix OnLine server.

(15) During an unplanned event such as a data center disaster, fast restores can be performed from the external backups. In an external backup, the SMV (Storage Management Vendor) is completely responsible for issuing all

I/O commands to move the data from disk to tape. In a normal DBMS managed backup, the database vendor issues the I/O command to read the data from disk, and the SMV then writes the data to tape. Conversely, in an external restore, the SMV issues all I/O commands from the tape back to disk. A DBMS managed restore has the SMV reading the data from tape, and the DBMS then writing the data to the disk. In both cases, data is restored from tape, and the roll forward (if required) with logical logs may occur.

(16) According to an illustrative embodiment of the present invention, there are two different types of external restore supported: complete external restore and partial external restore. In a complete external restore, all spaces will be restored to the most recent checkpoint that was generated while creating an external backup. A complete external restore is usually necessary when a major portion of the data storage system server needs to be restored or if an application error corrupts data. If users lose only a portion of the data (which is more typically the case), a partial external restore may be performed. A partial external restore restores only a subset of spaces that are down. This subset of spaces is defined by the user, with the limitation that it does not include any critical database spaces.

(17) After a complete external restore, the user can optionally do a logical restore to bring the server to logical consistency. A point-in-time restore can also be done as part of a complete external restore, to roll forward the logs to a specific time.

(18) According to an illustrative embodiment of the present invention the external backup is implemented by interfacing to the DBMS backup application to support querying, blocking, and unblocking the server (from access by the host systems). The physical external restore is done by the SMV software. The user then performs the logical restore by using extensions to the DBMS backup application.

(19) Advantages of the present invention include an external backup and restore mechanisms for a DBMS, such as Informix, where the mechanisms are effectively integrated with the DBMS. Other advantages include the seemless integration of external backup/restore procedures between the DBMS and the Storage Management Vendor (SMV). This invention automates the manual, and error prone external backup/restore procedures offered by the DBMS vendor.

BRIEF DESCRIPTION OF THE DRAWINGS

The foregoing and other features and advantages of the present invention will be more fully understood from the following detailed description of illustrative embodiments, taken in conjunction with the accompanying drawings in which:

FIG. 1 is a block diagram of a data storage system including backup components according to the prior art;

FIG. 2 is a block diagram showing the present invention integrated into the data storage system of FIG. 1; and

FIG. 3 is a flow chart of the data backup procedure according to an illustrative embodiment of the present invention.

(1) DETAILED DESCRIPTION

(2) An overview of major components of data system 10 is shown in FIG. 1. One or more host computer systems 12 access, process and store data from a data storage system 14. The host systems 12 are interfaced to the data storage system 14 over an interface 16, which may be any of various types of interface such as a Fibre or SCSI interface. The host systems 12 are also interfaced 20 to a backup system 22, which provides data backup and restore to appropriate storage devices 24, for example via tape storage. This interface 20 between the host systems 12 and the backup system 22 is also any of various types of interface, such as a TCP/IP connection.

(3) The data storage system 14 is any of various types of mass data storage systems, including for example a RAID system with multiple disks. A RAID-1

system is illustrated, with two **mirrored** disk volumes (**mirrors**) 18a, 18b. The **mirrors** 18a, 18b are connected 21 such that the data is replicated on both **mirrors** 18. Although the **mirrors** 18 are illustrated in a same data storage system 14 enclosure, the **mirrors** can be physically remote from each other, but still support RAID-1 **mirroring** using a remote data facility option, including a high-speed connection 21 such as an ESCON.RTM. fibre link connection.

(4) For backup and restore of data stored on the data storage system 14, a standard method for backup requires the host systems 12 to extract the data from the databases on the data storage system 14 and pipe the data over to the backup management system 22. This method is incredibly slow, and it requires tying up host systems 12 time in the form of database access operations and data pipelining. A better solution is known as "direct connect". A high speed direct connection 26 is provided between the data storage system 14 and the backup management system 22, thereby allowing fast pipelining of data directly to the backup management system 22, without the need for host system 12 intervention. This high speed direct connection 26 can be over any of various types of interfaces, such as a SCSI connection.

(5) An example data storage system 14 is the Symmetrix mass storage system provided by EMC Corporation of Hopkinton, Massachusetts. An example backup management system 22 is the EMC Data Manager (EDM). EDM can support backup and restore via three different methods, each tailored to particular backup environments and needs. The same EDM can support three different backup methods simultaneously.

(6) EDM runs a backup manager known as EDM Symmetrix Connect. EDM Symmetrix Connect is optimized for very large database (VLDB) environments, providing extremely high performance backup where the data movement between media (typically between disk and tape) is completely offloaded from the host systems 12 and the network. EDM exploits direct I/O capabilities to offer backup rates of hundreds of gigabytes per hour backup performance. Additionally, EDM offers high availability, using duplicate production volumes, either local or

remote
to run nondisruptive, point-in-time backups, and provides for nondisruptive
backup for UNIX and Windows NT. This enables host systems 12 and users to stay
operational and continue access to the data storage system 14 while backup
occurs.

(7) EDM also supports a direct connection 26 from the EDM backup client and
the data storage system 14 to the EDM system via an optimized path 26, offering
a fast NT **backup solution, while completely offloading the network 16 from the
backup** data stream. EDM supports large Oracle, Sybase, Informix or MS SQL
Server databases on popular UNIX and NT platforms. EDM supports backup of data
storage system-resident or local-disk-resident information using the data
storage system and the server to move the backup data over Ultra SCSI and Fibre
Channel connections from disk to tape at very high speeds. The control
data/handshaking is done over the **network 16 while the backup data is moved
over network** alternative data storage system channels 26.

(8) The data storage system 14 includes at least one database application,
which is accessed by the host systems 12 through the fast interface 16. Backup
of the database application over the high speed connection 26 avoids requiring
host system 12 intervention, however the advantage of the host system's
interface to the database is lost. For example, with an Oracle database on the
data storage system 14, the ability to use the host system's logical mapping to
the database is not possible. A solution to this problem is presented in U.S.
Pat. No. 6,047,294, issued Apr. 4, 2000, which is incorporated herein by
reference.

(9) The present invention is directed towards using the backup system 22FIG.
2 to perform external backup and restore of database spaces 30 in the data
storage system 14 by interfacing to the DBMS backup application 34 provided by
the database vendor. The database files are stored in database spaces 30 in
the data storage system 14, and are accessed by the host systems 12 through the
DBMS 32 running on the host systems 12. The backup system 22 controls

the external backup/restore. It interfaces to the DBMS backup application 34 for querying information about the database spaces 30, and for issuing commands to the DBMS backup application 34 for controlling the database spaces 30, such as blocking and unblocking host system access to the database spaces 30. The backup system 22 interfaces with the DBMS backup application 34 through the connection 20, but performs external backup/restore over the high speed direct connection 26. The logs are backed up by having the DBMS Backup Application 34 read the log files 30 and send the data over the interface 20 to the backup system 22 which then writes them to the tape media 24.

(10) An illustrative embodiment of the present invention provides an interface from an EMC EDM backup system 32 to an Informix DBMS. The Informix DBMS backup application 34 is the Informix Server which includes commands used for discovery, block and unblock. The illustrative embodiment also provides details of interfacing the EMC EDM backup system 32 to any vendor's DBMS backup application 34.

(11) The steps performed by the illustrative embodiment of the present invention are shown with. reference to FIG. 3. A backup process running on the backup system 32 commences with parsing any command line arguments provided, step 100. Next the system reads the configuration from the discovery data table (DDTAB) 40, step 102. The DDTAB is typically copied to the database host 12FIG. 2 for Discovery, Acquire and Release. The next step performed 104FIG. 3 depends on the particular backup phase, which includes Discover, Acquire, or Release. The particular backup phase is identified from the command line as parsed in step.

(12) A Discovery phase 105 is used to determine what components of a database are to be backed up (for example, a complete database backup of all files, or only selected files such as the files required for a backup of tables spaces). If individual table spaces are selected (through the command line), step 106, then the backup system 32 gets the file information for the selected spaces,

step 108. Spaces typically consist of multiple "chunks". These chunks describe a logical storage location on disk 30, FIG. 2. This storage location is the lowest level understanding of storage that the database application has. This information is stored in the DDTAB 40. The backup system uses this information in later phases of the discovery process to map this logical storage onto the exact physical storage locations on the disk 30 FIG. 2. When the appropriate application level files have been identified, a corresponding entry is made in the DDTAB 40.

(13) If all spaces have been selected at step 106 (a complete backup), then the system gets file information for all spaces in the server spaces 30, step 110. Corresponding entries are made in the DDTAB 40.

(14) Once the DDTAB 40 is properly. updated, the Discover phase returns successfully, step 126.

(15) The Acquire stage 107 is performed to block the DBMS server (32), thereby allowing the **mirror**-splitting process to take place. Steps 100-104 are again performed, which includes the system reading the configuration information from the DDTAB 40, which will indicate which spaces need to be backed up, as previously determined in the Discovery phase 105. The Acquire stage continues with the system checking to see if the DBMS server is blocked, step 110. If the DBMS server is already blocked then Acquire stage returns successfully, step 126.

(16) If the DBMS server is not already blocked, the system issues the commands to block the DBMS server, step 112. The system then checks to determine if the DBMS server was successfully blocked, step 114. A variety of DBMS error conditions might cause the block to be unsuccessful, for example a disk media failure can cause the flushing of data from cache memory to disk to fail, thereby making the block fail. If it was successfully blocked, then the Acquire stage returns successfully, step 126.

(17) However, if the attempt to block the DMBS server was not successful at step 114, the system then proceeds to issue commands to unblock the DBMS server

42, step 118. This is done prophylactically to ensure that the backup process never leaves the server in a persistently blocked state.

(18) The system then checks to see if the DBMS server was successfully unblocked, step 120. If the DBMS server was not successfully unblocked, then the Acquire stage returns unsuccessfully (returns with indications that the stage was unsuccessful), step 124. If the DBMS server was successfully unblocked at step 120, the Acquire stage still returns unsuccessfully at step 124, since the Acquire stage did not block the DBMS server 42, as required for the Acquire stage, (as indicated by step 122).

(19) Once the DBMS server is blocked, The EDM may acquire the disk resource (FIG. 214) by splitting **mirrors**. Once the **mirrors** have been split, system backup through Symmetrix connect occurs. The DDTAB file 40 is sent to the backup Symmetrix connect system. The DDTAB file 40 is used to determine what physical storage segments in the data storage system need to be backed up. Backup may take the form of first splitting the **mirrors** 18a, 18b and backing up the off-line **mirror** 18b. Similarly, the disks may be controlled and backed up as described in U.S. patent application Ser. No. 09/502,208 corresponding, entitled "System and Method for Backing Up Data Stored in a Mass Storage Subsystem Under Control of a Backup Server", filed on Mar. 31, 1998 which is incorporated herein by reference.

(20) When the all Acquire steps are complete, the Release stage is performed. Again Steps 100-104 are performed, which includes the system reading the configuration information from the DDTAB 40, which will indicate which spaces have been backed up and now need to be unblocked. The Release stage 109 next checks to see if the DBMS server is presently blocked, step 116. If the DBMS server is not presently blocked, then the Release stage returns unsuccessfully, step 124. If the server is not blocked at release time, it means that the **mirror** devices 30' FIG. 2 are not guaranteed to be suitable for backup. The server must be blocked at the time of the disk acquire phase. In

illustrative embodiment, the database is released back to the user after the **mirrors** have been split, but before the actual backup to tape. The illustrative embodiment can therefore detect this invalid condition before the movement of data begins, and fail the backup very early in the process.

(21) Otherwise, the system then proceeds to issue commands to unblock the DBMS server 42, step 118. The system then checks to see if the DBMS server was successfully unblocked, step 120. If the DBMS server was not successfully unblocked, then the Release stage returns unsuccessfully, step 124. If the DBMS server was successfully unblocked at step 120, the Release stage then returns successfully at step 126.

(22) Logs are not required for a full external restore, since a full external restore is consistent. Logs are required, however, for a partial external restore. It is required that the physically restored data be logically restored to make the server consistent. The user uses the standard vendor-specific DBMS backup utility functionality to ensure that logs are backed up. Typically, automatic backup log alarm archiving (to tape) is turned on. It is possible that any attempts by Symmetrix connect to affect the logs would interfere with this.

(23) The database layer code can be abstracted, so that Informix specific code can exist in its own module. This reduces the chances of regressions being introduced. In order to avoid introducing Informix specific code to the non-database layer Symmetrix connect code, dummy files and other constructs are utilized. For example, non-database layer code creates DO_FILE_LISTs for Oracle temporary control files, backs them up, and deletes them from disk. The Informix DBMS does not have this requirement. By having the Informix database layer create a dummy file for inclusion in the DDTAB, it is possible to avoid introducing Informix specific code. This type of approach also works for Oracle 8 Proxy Copy, and other DBMS implementations.

(24) For the illustrative embodiment interfaced to an Informix DBMS

to
support external backup/restore, the following modules are used with Symmetrix
Connect. Some modules are the same or similar to the external backup/restore
interface for Oracle DBMS as described in the referenced application.
eb_dc_db_itf: returns the database interface type. Some example return values
include Oracle, Informix, SAP/R3, Backint Interface (this implies Oracle is the
database), and MS SQLServer. The present invention will work with any new
database interface type which are developed. This function calls the DBMS or
proxy specific interface layers. In the illustrative embodiment, the database
type would be read from the DDTAB as "informix", and eb_dc_inf_itf is then
invoked. eb_dc_inf_itf: This module has the Informix specific database
interface code. The database interface layer uses eb_exec_as to execute
Informix commands. For Unix systems, this allows the root user on the database
machine to run commands as the Informix User.

(25) The new operative command for Informix Discovery is: select*
a.name,b.fname from sysdbspaces a,syschunks b where a.dbsnum=b.dbsnum
onstat -d
may all be used in some circumstances, as well as consulting the ONCONFIG
configuration file.

(26) The operative command for Informix preparation is: onmode-c block

(27) The operative command for Informix release is: onmode-c unblock
eb_dc_config: configuration module, that now is sensitive to the database type,
and modified to handle the different terms used by Informix for backup objects.
For example, Informix DBMS uses terms such as Server name, dbspace and
blobspace. As interfaces to other DBMS implementations may be added, the
implementation specific config components can optimally be changed to function
calls that are sourced in, based on the database type. eb_dc_restore:
restoration module, similar to eb_dc_config is modified to handle interfaces to
other DBMS implementations.

(28) The modules discussed here are the ones that are changed or be added
within the current Symmetrix Connect product according to the illustrative
embodiment. They do not represent a complete list of all modules that are
invoked to run a backup or restore. Further, another embodiment

involves
changing the core Symmetrix Connect product such that configuration (eb_dc_config) is all be done via the EDM GUI (Graphic User Interface) application.

(29) Although the invention has been shown and described with respect to illustrative embodiments thereof, various other changes, omissions and additions in the form and detail thereof may be made therein without departing from the spirit and scope of the invention.

CLAIMS:

What is claimed is:

1. A method for backing up data stored in data storage system, said data controlled by a DBMS (data base management system) running on a host computer connected to said data storage system, said method comprising: interfacing to said DBMS for said data; querying said DBMS regarding said data, to determine files to be backed up; commanding said DBMS to block access to said data by said host computer; performing a backup of said files to be backed up without using said DBMS; commanding said DBMS to unblock access to said data by said host computer.

2. The method of claim 1 further including: after said step of commanding said DBMS to block access to said data by said host computer, checking with said DBMS to confirm access to said data has been blocked.

3. The method of claim 1 further including: after said step of commanding said DBMS to unblock access to said data by said host computer, checking with said DBMS to confirm access to said data has been unblocked.

4. The method of claim 1 wherein said DBMS is an Informix database.

5. A method for backing up data stored in data storage system, said data controlled by a DBMS (data base management system) running on a host computer connected to said data storage system, said method comprising: interfacing to said DBMS for said data; querying said DBMS regarding said data, to determine files to be backed up; commanding said DBMS to block access to said data by said host computer; performing a backup of said files to be backed up

without
using said DBMS; commanding said DBMS to unblock access to said data by said
host computer; wherein said data storage system includes a RAID-1 data storage
system with **mirrored** storage devices, and said step of performing a backup
includes splitting apart said **mirrored** storage devices.

6. The method of claim 5 wherein said step of performing a backup includes
after splitting apart said **mirrored** storage device, commanding said DBMS to
unblock access to said data by said host computer, and performing a backup on
an offline **mirror of said mirrored** storage devices.

7. A system for backing up data stored in data storage system, said data
controlled by a DBMS (data base management system) running on a host computer
connected to said data storage system, said method comprising: a backup system,
including at least one backup storage device, and in communication with said
host computer and also in communication with said data storage system, wherein
said backup system performs the steps of: interfacing to said DBMS running on
said host computer; querying said DBMS regarding said data, to determine files
to be backed up from said data storage system; commanding said DBMS to block
access to said data by said host computer; performing a backup of said files
to be backed up to said at least one backup storage device, without using said
DBMS; commanding said DBMS to unblock access to said data by said host computer.

8. A system for backing up data stored in data storage system, said data
controlled by a DBMS (data base management system) running on a host computer
connected to said data storage system, said method comprising: a backup system,
including at least one backup storage device, and in communication with said
host computer and also in communication with said data storage system, wherein
said backup system performs the steps of: interfacing to said DBMS running on
said host computer; querying said DBMS regarding said data, to determine files
to be backed up from said data storage system; commanding said DBMS to block

access to said data by said host computer; performing a backup of said files
to be backed up to said at least one backup storage device, without using said
DBMS; commanding said DBMS to unblock access to said data by said host computer; wherein said data storage system includes a RAID-1 data storage
system with **mirrored** storage devices, and said backup system splits apart said
mirrored storage devices, host system, and performing a backup on an offline
mirror of said mirrored storage device.

L Number	Hits	Search Text	DB	Time stamp
11	14995	(backup or back$4) near4 network	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/03/14 13:47
12	1235	target adj2 node	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/03/14 13:47
13	9100	source adj2 node	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/03/14 13:48
14	474026	mirror$4	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/03/14 13:48
15	1118	logical adj2 volum$2	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/03/14 13:48
16	6114	plex$3	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/03/14 13:48
17	412	(target adj2 node) and (source adj2 node)	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/03/14 13:48
18	1259	((backup or back$4) near4 network) and mirror$4	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/03/14 13:49
20	1	((((backup or back$4) near4 network) and mirror$4) and (logical adj2 volum$2)) and plex$3	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/03/14 13:50
21	1	((((backup or back$4) near4 network) and mirror$4) and (logical adj2 volum$2)) and ((target adj2 node) and (source adj2 node))	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/03/14 13:50

19	50	(((backup or back$4) near4 network) and mirror$4) and (logical adj2 volum$2)	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/03/14 13:50